FOR MORE INFORMATION:

NASA-STD-8719.13B
NASA Software Safety Standard

NASA-GB-8719.13
NASA Software Safety Guidebook

Both available at:
http://www.hq.nasa.gov/office/codeq/doctree/safeheal.htm

Software Safety Training:
https://solar.msfc.nasa.gov

Other Software Safety Information:
http://standards.nasa.gov
http://nodis3.gsfc.nasa.gov
http://www.hq.nasa.gov/office/codeq/software

YOUR CENTER SOFTWARE SAFETY REPRESENTATIVE:

10 STEPS TO SAFER SOFTWARE

1. Evaluate your system and software for hazards
2. Plan for software safety activities and analyses
3. Follow sound Software Engineering and Software Assurance processes

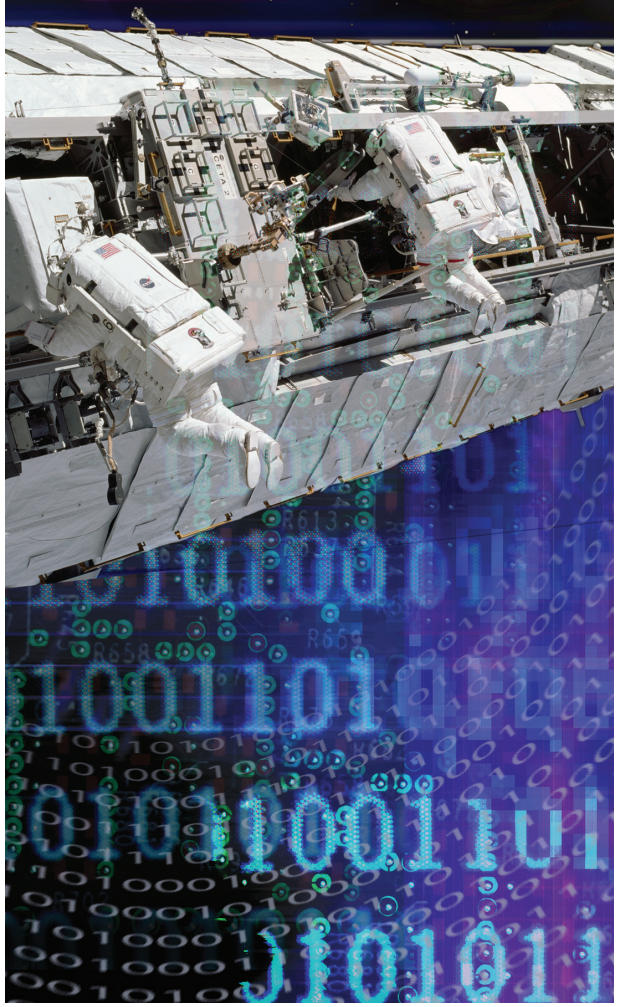
4. Review, analyze, document, and verify
5. Communicate and coordinate
6. Manage and trace software safety through the entire life cycle

7. Evaluate off-the-shelf and reused software for your system
8. Certify your system according to your Center's process
9. Be creatively paranoid! Always think how the system can fail
10. Remember: Software safety doesn't stop at system delivery!

NP-2004-07-366-HQ

NASA Software Safety

Rising to New Heights



An Overview of NASA Software Safety

Don't Panic!

While the requirements for software safety apply to all projects with safety critical software, the implementation of those requirements can vary tremendously!

One size does not fit all,
we can help tailor to your needs.

EVALUATE YOUR SYSTEM

Step 1: Perform System Safety Analyses to determine if your system is safety-critical

Step 2: Your software is safety-critical if it meets at least one of these conditions:

1. Resides in a safety-critical system AND:
   a. Causes or contributes to a hazard
   b. Controls or mitigates hazards
   c. Controls safety-critical functions.
   d. Processes safety-critical commands or data
   e. Detects, reports, or takes corrective action, if the system reaches a specific hazardous state
   f. Mitigates damage if a hazard occurs
   g. Resides on the same system (processor) as safety-critical software
2. Processes data or analyzes trends that lead directly to safety decisions
3. Provides full or partial verification or validation of safety-critical systems

Step 3: Follow requirements in NASA-STD-8719.13

Step 4: Get scoping and tailoring help, and checklists and examples from the NASA Software Safety Guidebook (NASA-GB-8719.13)

SOFTWARE SAFETY INVOLVES:

1. Integrating safety into the software life cycle
2. Analyzing the software, system, and interfaces from beginning to end
3. Documenting safety plans, decisions, processes, and results
4. Using trained software safety personnel to help scope and tailor efforts for YOUR project
5. Tracing software safety requirements through all software phases
6. Reporting and resolving problems and discrepancies
7. Controlling software configuration
8. Evaluating off-the-shelf software
9. Managing contractors
10. Certifying the software

DEVELOP SAFER SOFTWARE

1. Create a complete set of software safety requirements that provide adequate response and system protection
2. Incorporate all software safety requirements in the software design and code
3. Verify that the design does not compromise safety controls or create additional hazards
4. Create and follow good coding standards
5. Verify that the software code reflects the safety requirements
6. Test the software at the unit, component, and system level under load, stress, and simulated failures
7. Document, document, document!

SOFTWARE SAFETY CONTINUES DURING OPERATIONS

1. Software safety applies to a system until it is retired
2. Software upgrades, updates, fixes, and other changes must be analyzed for safety impacts
3. User manuals must describe safety-related commands and data.